Mapped polynomials and discontinuous kernels for Runge and Gibbs phenomena

Stefano De Marchi

Abstract In this paper, we present recent solutions to the problem of approximating functions by polynomials for reducing in a substantial manner two well-known phenomena: Runge and Gibbs. The main idea is to avoid resampling the function or data and relies on the *mapped polynomials* or "fake" nodes approach. This technique turns out to be effective for stability by reducing the Runge effect and also in the presence of discontinuities by almost cancelling the Gibbs phenomenon. The technique is very general and can be applied to any approximant of the underlying function to be reconstructed: polynomials, rational functions or any other basis. A "natural" application is then quadrature, that we started recently to investigate and we propose here some results. In the case of jumps or discontinuities, where the Gibbs phenomenon appears, we propose a different approach inspired by approximating functions by kernels, in particular Radial Basis Functions (RBF). We use the so called *Variably Scaled Discontinuous Kernels (VSDK)* as an extension of the Variably Scaled Kernels (VSK) firstly introduced in [18]. VSDK show to be a very flexible tool suitable to substantially reducing the Gibbs phenomenon in reconstructing functions with jumps [28, 55]. As an interesting application we apply VSDK in *Magnetic Particle Imaging* which is a recent non-invasive tomographic technique that detects super-paramagnetic nanoparticle tracers and finds applications in diagnostic imaging and material science [39, 47]. In fact, the image generated by the MPI scanners are usually discontinuous and sampled at scattered data points, making the reconstruction problem affected by the Gibbs phenomenon. We show that VSDK are well suited in MPI image reconstruction also for identifying image discontinuities [27].

Stefano De Marchi

Department of Mathematics "Tullio Levi-Civita", Via Trieste 63 - 35121 Padova (Italy), e-mail: demarchi@math.unipd.it

1 Introduction

Univariate approximation of functions and data is a well studied topic since the antiquity (Babylon and Greece), with many different developments by the Arabs and Persians in pre and late medieval period. The scientific revolution in astronomy, mainly due to Copernicus, Kepler and Galileo was the starting point for the investigations done later on by Newton, who gave strong impetus to the further advancement of mathematics, including what is now called "classical" interpolation theory (interested people on these historical aspects may read the nice survey by E. Meijering [50]).

Interpolation is essentially a way of estimating a given function $f : [a, b] \subset \mathbb{R} \to \mathbb{R}$ known only at a finite set $X_n = \{x_i, \ i = 0, \ldots, n\}$ of $n + 1$ (*distinct*) points, called *interpolation points* The corresponding set of values is denoted by $Y_n = \{y_i = f(x_i), \ i = 0, \ldots, n\}$. Then, we are looking for a polynomial $p \in \mathbb{P}_n$, with $\mathbb{P}_n$ the linear space of polynomials of degree less and equal than n. The space $\mathbb{P}_n$ has finite dimension $n + 1$ and spanned by the monomial basis $M = \{1, x, x^2, \ldots, x^n\}$. Therefore, every $p \in \mathbb{P}_n$ is written as

$$p(x) = \sum_{j=0}^{n} a_j x^j \, . \tag{1}$$

The coefficients a_i are determined by solving the linear system $p(x_i) = y_i$, $i = 0, \ldots, n$. Introducing the *Vandermonde* matrix $V_{ij} = (x_i^j)$, the coefficient vector $\boldsymbol{a} = (a_0, \ldots, a_n)^t$ and the vector $\boldsymbol{y} = (y_0, \ldots, y_n)^t$, the linear system can compactly be written as

$$V\boldsymbol{a} = \boldsymbol{y} \, . \tag{2}$$

As well-known, the solution of the system is guaranteed if the points are *distinct*, making V invertible [21, §2]. Moreover, the interpolating polynomial (1) can be expressed at any point $x \in [a, b]$ by the discrete inner product $p = \langle \boldsymbol{a}, \boldsymbol{x} \rangle$.

Instead of the monomial basis, we can use the cardinal basis of *elementary Lagrange polynomials*. That is $L = \{\ell_i, \ i = 0, \ldots, n\}$ where $l_i(x) = \prod_{j=0, j \neq i} \dfrac{x - x_j}{x_i - x_j}$

or, alternatively by the the ratio

$$\ell_i(x) = \frac{\det V_i(x)}{\det V} \tag{3}$$

where, $V_i(x)$ is the Vandermonde matrix in which we have substituted the i-th column with the vector $\boldsymbol{x} = (1, x, x^2, \ldots, x^n)^T$. The formula (3) is essentially the Cramer's rule applied to the system

$$V\boldsymbol{\ell} = \boldsymbol{x}$$

showing immediately the main property of the elementary Lagrange polynomials: they form a set of cardinal polynomial functions, that is $\ell_i(x_j) = \delta_{ij}$. Using the Lagrange basis the interpolant becomes $p = \langle \boldsymbol{l}, \boldsymbol{y} \rangle$. Therefore, by unicity of interpo-

lation we get

$$\langle a, x \rangle = \langle y, \ell \rangle .$$

Hence, in the Lagrange basis ℓ the vector of coefficients a is at once y, so that in (2) V is substituted by the identity matrix I of order $n + 1$.

There is another interesting formula that we can used to express the interpolant p. As pointed out in [20, §3 Prob. 14], the interpolant at every point x can be written in determinantal form as follows

$$p(x) = C \cdot \det \begin{bmatrix} 0 & 1 & x & \cdots & x^n \\ \hline y_0 & & & & \\ y_1 & & V & & \\ \vdots & & & & \\ y_n & & & & \end{bmatrix} . \qquad (4)$$

This expresses the interpolant as the determinant of an $(n + 2) \times (n + 2)$ matrix, obtained by bordering the Vandermonde matrix with two vectors y, x^T and the scalar 0. The constant C appearing is (4) is a normalizing factor for expressing the interpolant in Lagrange form, that is $C = -1/\det(V)$.

This formula can be specialized for any set of linear independent functions, say $\{g_0, \ldots, g_n\}$ (cf. [20, §3, Prob. 15]) and in particular for the Lagrange basis L obtaining

$$p(x) = -\det \begin{bmatrix} 0 & \ell_0(x) & \ell_1(x) & \cdots & \ell_n(x) \\ \hline y_0 & & & & \\ y_1 & & I & & \\ \vdots & & & & \\ y_n & & & & \end{bmatrix} , \qquad (5)$$

with I the identity matrix of order $n + 1$.

Historically, but also nowadays in different frameworks and applications, the simplest way to take distinct samples x_i, is to consider equally spaced points (assuming for simplicity $x_0 = a$ and $x_n = b$). Two well-known phenomena are related to this choice of the interpolation points. The first one is the *Runge phenomenon*: when using polynomial interpolation with polynomials of high degree, the polynomial shows high oscillations at the edges of the interval. It was discovered by Carl Runge when exploring the behavior of errors when using polynomial interpolation to approximate also analytic functions (cf. [56]). It is related to the *stability* of the interpolation process via the *Lebesgue constant*

$$\Lambda_n := \max_{x \in [a,b]} \sum_{i=0}^{n} |l_i(x)| \qquad (6)$$

by means of the inequality

$$\|f - p\|_\infty \le (1 + \Lambda_n)\|f - p^*\|_\infty \qquad (7)$$

where p^* represents the polynomial of best uniform approximation, that is $p^* := \inf_{p \in \mathbb{P}_n} \|f - p\|_\infty$, which uniquely exists. About the growth of the Lebesgue constant and its relevant properties we invite the readers to refer to the great survey by L. Brutman [19]. Here we simply recall the fundamental fact that Λ_n, which depends only on the choice of the node set X, grows exponentially (with n) when the interpolation points are equally spaced. Therefore it will be fundamental to look for a different choice than the equal distribution. As well-known, this is what do the *Chebyshev points* in $[a, b]$ or the *zeros of orthogonal polynomials* with respect to the interval $[a, b]$ and a given weight function (cf. e.g. [19, 21]).

The second phenomenon and related to the Runge phenomenon is the *Gibbs phenomenon*, which tells us the peculiar manner in which the Fourier series of a piecewise continuously differentiable periodic function behaves at a jump discontinuity. This effect was originally discovered by Henry Wilbraham (1848) and successively rediscovered by J. Willard Gibbs (1899) (see [42]). The phenomenon observed that the n-th partial sum of the Fourier series has large oscillations near the jump, which might increase the maximum of the partial sum above that of the function itself. The overshoot does not die out as n increases, but approaches a finite limit. The Gibbs phenomenon is the cause of unpleasant artifacts in signal and image processing in presence of discontinuities.

The Gibbs phenomenon is also a well-known issue in higher dimensions and for other basis systems like wavelets or splines (cf. e.g. [44] for a general overview) and also in barycentric rational approximation [51]. Further, it appears also in the context of Radial Basis Function (RBF) interpolation [36]. To soften the effects of this artifact, additional smoothing filters are usually applied to the interpolant. For radial basis function methods, one can for instance use linear RBFs in regions around discontinuities [45, 25]. Furthermore, post-processing techniques, such as Gegenbauer reconstruction procedure [40] or digital total variation [57], are available. This technique can be also applied in the non-polynomial setting by means of *discontinuous kernels*.

This survey paper consists of 6 main sections and various subsections as follows. In Section 2 we introduce our main idea for mitigating the Runge and Gibbs effects based on the mapped-basis approach or its equivalent form that we have termed "fake-nodes". In the successive Section 3 we present the algorithms for choosing a suitable map in the two instances studied in the paper. In Sections 4 we show that the technique can be applied also using rational approximation instead of polynomial, while in Section 5 we discuss the application to quadrature. We continue then with Section 6 in which, for treating the Gibbs phenomenon we use a non-polynomial approach based on discontinuous kernels, in particular the so-called VSDK. Finally in Section 7 we discuss the application of these VSDK to the Magnetic Particle Imaging, a new

quantitative imaging method for medical applications. We conclude with Section 8 by summarizing the main ideas and highlighting some further developments.

2 Mitigating the Runge and Gibbs phenomena

Let $[a, b] \subset \mathbb{R}$ be an interval, X the set of distinct nodes (also called data sites) and $f : \Omega \longrightarrow \mathbb{R}$ a given function sampled at the nodes with $F_n = \{f_i = f(x_i)\}_{i=0,\ldots,n}$.

We now introduce a method that changes the interpolation problem (2) without resampling the function f. The idea rely on the so-called *mapped basis approach* where the map is used to mitigate the oscillations in the Runge and Gibbs phenomena. The idea of mapped polynomials is not new. Indeed, such kinds of methods have been used in the context of spectral schemes for PDEs.

For examples of well-known maps refer to e.g. [41, 49]. However, for our purposes, that are devoted especially to applications when resampling cannot be performed, we consider a generic map $S : [a, b] \longrightarrow \mathbb{R}$. We investigate the conditions which this map S should satisfy in Subsection 2.1.

Let $\hat{x} \in \hat{S} := S([a, b])$ we can compute the polynomial $P_g : \hat{S} \longrightarrow \mathbb{R}$ interpolating the function values F_n at the *"fake" nodes* $S_X = \{S(x_i) = \hat{x}_i\}_{i=0,\ldots,n} \subset \hat{S}$ defined by

$$P_g(\hat{x}) = \sum_{i=0}^{n} c_i \, \hat{x}^i,$$

for some smooth function $g : \hat{S} \longrightarrow \mathbb{R}$ such that

$$g_{|S_X} = f_{|X_n}.$$

Hence, for $x \in [a, b]$ we are interested in studying the function

$$R_f^S(x) := P_g(S(x)) = \sum_{i=0}^{n} c_i \, S(x)^i. \tag{8}$$

The function R_f^S in (8) can be considered as an interpolating function at the original set of nodes X_n and data values F_n, which is a linear combination of the basis functions $\mathcal{S}_n := \{S(x)^i, \ i = 0, \ldots, n\}$. As far as we know, a similar approach has been mentioned in [5], without being later worked out.

The analysis of this interpolation process can be performed in the following equivalent ways.

- The *mapped bases* approach on $[a, b]$: interpolate f on the set X_n via R_f^S in the function space $\mathcal{S}_n$.
- The *"fake" nodes* approach on $\hat{S}$: we interpolate g on the set S_X via P_g in the polynomial space Π_n.

2.1 The mapped bases approach

The first question to answer is: how arbitrary is the choice of the map S?

Definition 1 We term S *admissible* if the resulting interpolation process admits a unique solution, that is the Vandermonde-like matrix $V^S := V(S_0, \ldots, S_n)$, is invertible.

Since the new point set in the mapped space is S_X, then

$$\det(V^S) = \prod_{0 \le i < j \le n} (S_j - S_i) \neq 0.$$

A consequence of this observation is the following proposition.

Proposition 1 *The map S is admissible if and only if for any $0 \le i, j \le n$, $i \neq j$ we have $S_i \neq S_j$. In other words, S is injective in $[a, b]$.*

In fact, $\det(V^S) \neq 0$ if and only if $S_j - S_i \neq 0$.

Remark 1 We point out that we can easily write

$$\det(V^S) = \sigma(S, X)\det(V),$$

where V is the classical Vandermonde matrix and

$$\sigma(S, X) := \prod_{0 \le i < j \le n} \frac{S_j - S_i}{x_j - x_i}.$$

This fact presents some similarities with the so-called *generalized Vandermonde determinants* that can be factored as the classical Vandermonde determinant and a Schur function, as outlined for instance in the paper [22].

We can associate to the interpolant R_f^S also its Lagrange form

$$R_f^S(x) = \sum_{i=0}^{n} f_i \ell_i^S(x),$$

with, in analogy with (3),

$$\ell_i^S(x) := \frac{\det(V_i^S(x))}{\det(V^S)} = \prod_{\substack{0 \le j \le n \\ j \neq i}} \frac{S(x) - S_j}{S_i - S_j}, \tag{9}$$

where $V_i^S(x) = V_i(S(x))$.

Consequently, we can consider the S-Lebesgue constant Λ_n^S associated to the mapped Lagrange basis $\mathcal{L}^S = \{\ell_0^S, \ldots, \ell_n^S\}$ and to the interpolation operator $\mathcal{L}_n^S$:

$\Upsilon \longrightarrow S_n$ with $(\Upsilon, \|\cdot\|_\Omega)$ be a normed function space, which contains only real-valued functions on $\Omega = [a, b]$. Then, Λ_n^S is the operator norm of $\mathcal{L}_n^S$ with respect to $\|\cdot\|_\Omega$, that is

$$\Lambda_n^S = \sup_{\substack{v \in \Upsilon \\ v \neq 0}} \frac{\|\mathcal{L}_n^s(v)\|_\Omega}{\|v\|_\Omega},$$

and hence we can provide the sup-norm error estimator as follows:

$$\|f - R_f^S\|_\Omega \leq (1 + \Lambda_n^s(\Omega))E_n^{S,\star}(f),$$

where $E_n^{S,\star}(f)$ is the best polynomial approximation error in the sup-norm [53, Theorem I.1, p. 1].

We have just seen that ℓ_i and ℓ_i^S are defined as in 3 and (9), respectively. The next Proposition proposes a rough upper bound for the S-Lebesgue constant, useful for understanding the asymptotic behaviour of the error.

Proposition 2 *For all $x \in [a.b]$, $x \neq x_j$, we have*

$$\ell_i^S(x) = \underbrace{\frac{\beta_i(x)}{\alpha_i}}_{\gamma_i(x)} \ell_i(x), \tag{10}$$

with $\alpha_i := \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{S_i - S_j}{x_i - x_j}, \quad \beta_i(x) := \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{S(x) - S_j}{x - x_j}.$

Proof By construction $\ell_i^S(x_j) = \ell_i(x_j) = \delta_{i,j}$. Then, let $x \neq x_j$, from (3) and (9) we have that

$$\frac{\ell_i^S(x)}{\ell_i(x)} = \frac{\det(V_i^S(x))\det(V)}{\det(V_i(x))\det(V^s)} = \frac{\det(V_i^S(x))}{\sigma(S, X)\det(V_i(x))} := \gamma_i(x).$$

We can also write

$$\ell_i^S(x) = \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{S(x) - S_j}{S_i - S_j} = \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - x_j}{x_i - x_j} \cdot \frac{x_i - x_j}{x - x_j} \cdot \frac{S(x) - S_j}{S_i - S_j},$$

$$= \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - x_j}{x_i - x_j} \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{S(x) - S_j}{x - x_j} \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x_i - x_j}{S_i - S_j}.$$

By defining

$$\alpha_i := \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{S_i - S_j}{x_i - x_j}, \quad \beta_i(x) := \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{S(x) - S_j}{x - x_j},$$

we get formula (10) as claimed. $\square$

As a consequence, we can bound Λ_n^S from above by Λ_n unless a constant $C(S, n)$ depending on the map S and n.

Theorem 1 *Letting*

$$L = \max_{0 \leq i \leq n} \max_{x \in [a,b], i \neq j} \left| \frac{S(x) - S_j}{x - x_j} \right|, \quad D = \min_{0 \leq i \leq n} \min_{j \neq i} \left| \frac{S_i - S_j}{x_i - x_j} \right|,$$

then

$$\Lambda_n^S \leq C(S, n)\Lambda_n, \tag{11}$$

where $C = (L/D)^n$ with Λ_n the classical Lebesgue constant in (6).

Proof Using Proposition 2

$$|\ell_i^S(x)| = \left| \frac{\beta_i(x)}{\alpha_i} \right| |\ell_i(x)|.$$

An upper bound for $|\beta_i|$ is

$$|\beta_i(x)| = \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \left| \frac{S(x) - S_j}{x - x_j} \right| \leq \prod_{\substack{0 \leq j \leq n \\ j \neq i}} L_i,$$

where

$$L_i = \max_{x \in [a,b], i \neq j} \left| \frac{S(x) - S_j}{x - x_j} \right|.$$

Thus,

$$|\beta_i(x)| \leq L_i^n.$$

A lower bound for $|\alpha_i|$ is

$$|\alpha_i| = \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \left| \frac{S_i - S_j}{x_i - x_j} \right| \geq \prod_{\substack{0 \leq j \leq n \\ j \neq i}} D_i = D_i^n,$$

where

$$D_i := \min_{j \neq i} \left| \frac{S_i - S_j}{x_i - x_j} \right|.$$

We then have

$$|\ell_i^s(x)| \leq \left(\frac{L_i}{D_i} \right)^n |\ell_i(x)|.$$

Therefore, letting $L := \max_{i \neq j} L_i$, $D := \min_i D_i$ and considering the sum of the Lagrange polynomials, we obtain

$$\Lambda_n^s \leq \left(\frac{L}{D} \right)^n \Lambda_n.$$

We conclude by setting $C(S, n) = (L/D)^n$. $\square$

2.2 The "fake" nodes approach

The construction of the interpolating function R_f^S is equivalent to building a polynomial interpolant at the "fake" nodes, as defined in (8). Therefore, in what follows we concisely analyze the parallelism with the polynomial interpolation problem in $\hat{S}$.

If ℓ_i is the i-th Lagrange polynomial related to the set S_X, then for $\hat{x} \in \hat{S}$, we have

$$\ell_i(\hat{x}) = \prod_{\substack{0 \le j \le n \\ j \ne i}} \frac{\hat{x} - S(x_j)}{S(x_i) - S(x_j)},$$

and the Lebesgue constant is given by

$$\Lambda_n(S(\Omega)) = \max_{\hat{x} \in S(\Omega)} \sum_{i=0}^{n} |\ell_i(\hat{x})|. \tag{12}$$

For $x \in \Omega$, we observe that

$$\ell_i(\hat{x}) = \ell_i(S(x)) = \prod_{\substack{0 \le j \le n \\ j \ne i}} \frac{S(x) - S(x_j)}{S(x_i) - S(x_j)} = \ell_i^s(x).$$

As a consequence, we obtain

$$\Lambda_n^S(\Omega) = \Lambda_n(S(\Omega)),$$

and

$$\|f - R_f^S\|_\Omega = \|g - P_{n,g}\|_{S(\Omega)},$$

which in particular implies that

$$\|f - R_f^S\|_\Omega \le (1 + \Lambda_n(S(\Omega)))E_n^\star(g).$$

Since we can suppose without loss of generality that g is a regular function, for an appropriate choice of the map S, and hence of the nodes S_X, we might improve the results with respect to classical polynomial approximation in $[a, b]$. The main difficulties are in finding a *good* map. In the next section we thus propose two receipts for computing suitable maps that, as numerically shown later, enable us to naturally mitigate both Runge and Gibbs phenomena.

3 Choosing the map S: two algorithms

In this Section, we describe how, given an *ordered set* of interpolation nodes $X_n = \{x_i \in [a, b] \mid x_0 = a, \ x_n = b, \ x_i < x_{i+1}\}$, we can effectively construct suitable maps S. We present two different ways of constructing the map S and in doing so we deal with the Runge and Gibbs phenomenon, respectively.

Treating the Runge phenomenon. In order to prevent the appearance of the Runge phenomenon, we construct a map S such that the resulting set of "fake" nodes S_X guarantees a stable interpolation process. The natural way is mapping X_n to the set of ordered Chebyshev-Lobatto (CL) nodes $C_n = \{c_0, \dots, c_n\}$ on $[a, b]$.

Indeed, as well-known the Lebesgue constant of the CL nodes grows logarithmically with respect to n [19]. The map S on $[a, b]$ that does this transformation, for any $x \in [a, b]$, is the piecewise interpolant as obtained by the following algorithm that we term *S-Runge*.

Algorithm 1

Input: X_n.

1. Define the set of CL nodes C_n.
2. For $x \in [x_i, x_{i+1}]$, $i = 0, \dots, n - 1$, set $\beta_{1,i} = \frac{c_{i+1} - c_i}{x_{i+1} - x_i}$, $\quad \beta_{2,i} = c_i$ so that $S(x) = \beta_{1,i}(x - x_i) + \beta_{2,i}$.

Outputs: the vectors $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2$ of the coefficients that define S.

Since the CL nodes are distinct, the map is admissible by construction. For instance, if $X_n = \{x_k = a + k \cdot (b - a)/n, k = 0, \dots, n\}$ is the ordered set of $n + 1$ equispaced nodes in $[a, b]$, it can be analytically mapped to C_n by using the map

$$S(x) = \frac{a - b}{2} \cos\left(\pi \frac{x - a}{b - a}\right) + \frac{a + b}{2}. \tag{13}$$

This algorithm, is robust and does not require any additional hypothesis on X_n and it could work for scattered data or on random, nearly equispaced or *well-spaced* interpolation nodes (for the definition of well-spaced we refer to [16]). These algorithms are also quoted in Wikipedia at [61].

Treating the Gibbs phenomenon. Let us suppose that the underlying function f presents jump discontinuities, whose positions and magnitudes are encoded in the set

$$D_m := \{(\xi_i, d_i) \mid \xi_i \in (a, b), \ \xi_i < \xi_{i+1}, \ i = 0, \dots, m, \ \text{and} \ d_i := |f(\xi_i^+) - f(\xi_i^-)|\}.$$

We assume to know the discontinuities and the related jumps. Such assumption is not restrictive. Indeed for the one dimensional case, but also in higher dimensions,

algorithms for detecting the discontinuity points are available, see for instance the references [4, 54].

Near a discontinuity, the interpolant is forced to strongly vary making the Gibbs phenomenon more evident. The usual approach is to put more points nearby the jumps or use a so-called filter in order to get more accurate solution than rough Fourier expansions (cf e.g. [25]) or acceleration techniques of the Fourier expansions based on the ϵ-algorithm (see e.g. [6]).

Our strategy is very simple: we sufficiently increase the gap between the node right before and the one right after the discontinuity, so allowing the interpolant to become smoother. To accomplish this, we introduce a *shifting parameter $k > 0$*.

The next algorithm, that we call *S-Gibbs* implements this idea.

Algorithm 2

Inputs: D_m and k.

1. Letting $A_i = \displaystyle\sum_{j=0}^{i} k d_j, \ i = 0, \ldots, m$.

2. Define S as

$$S(x) = \begin{cases} x, & \text{for } x \in [a, \xi_0[, \\ x + A_i, & \text{for } x \in [\xi_i, \xi_{i+1}[, \ 0 \le i < m, \text{ or } x \in [\xi_m, b]. \end{cases}$$

Outputs: the values A_i.

As experimentally observed the choice of k is not critical. It suffices that in the mapped space the so-constructed function g has no steep gradients and this can be obtained by taking $k \gg 1$. Since the resulting "fake" nodes S_X are distinct, the so constructed map is admissible.

3.1 Numerical tests

We show via the algorithms described in the previous Section 3 that we are able to substantially reduce the oscillating effects due to Runge and Gibbs phenomena. Our "fake" nodes approach is compared with the resampling at Chebyshev nodes. We note, that in many applications we unfortunately dispose of the data at equispaced samples. It is the reason why our approach becomes relevant for the underlying idea: *mapping without resampling*.

We consider the interval $[-5, 5]$ and both equispaced and randomly distributed intepolation nodes. Moreover, we evaluate the interpolants on a set of equispaced points $\Xi = \{\bar{x}_i, i = 0, \ldots, 330\}$ and compute the Relative Maximum Absolute Error (RMAE)

$$\text{RMAE} = \max_{i=0,\ldots,m} \frac{|R_{n,f}^s(\bar{x}_i) - f(\bar{x}_i)|}{|f(\bar{x}_i)|}. \tag{14}$$

The experiments have been carried out in PYTHON 3.6 using Numpy 1.15; see [29].

3.1.1 Application to Runge phenomenon

For this test we consider the function $f_1(x) = \frac{1}{e^{-3x}+1}$ sampled at equispaced interpolation nodes E_n. We then compare f_1 evaluated at Ξ with respect to

   i. the interpolating polynomial at equispaced points E_n, i.e the original data set and function values;

   ii. the interpolating polynomial at CL nodes C_n and resampled function values $f_1(C_n)$, i.e. we resample the function;

   iii. the approximant built upon a polynomial interpolant at the fake nodes $S(E_n)$ corresponding to the CL, C_n obtained by the map (13), and function values $f_1(E_n)$.

In Fig. 1, we show different reconstructions of f_1 for a fixed number of nodes. In Fig. 2 we show the RMAE for the function f_1 varying the number of nodes while in Fig. 3 we plot the Lebesgue functions related to the proposed methods. As pointed out in the theoretical analysis, the behavior of the "fake" nodes in terms of Lebesgue constant is analogous to that of the classical polynomial interpolation at CL points.
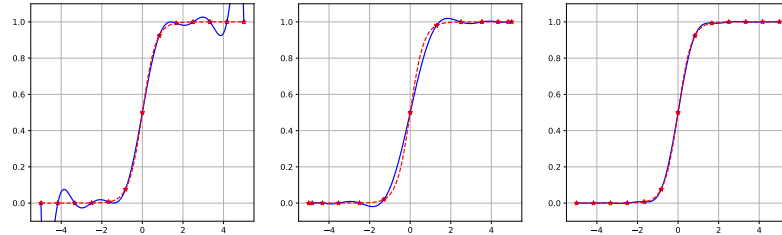

Fig. 1 Interpolation with 13 points of the function f_1 on $[-5, 5]$ using equispaced (left), CL (center) and "fake" nodes (right). The nodes are represented by stars, the original and reconstructed functions are plotted with continuous red and dotted blue lines, respectively.

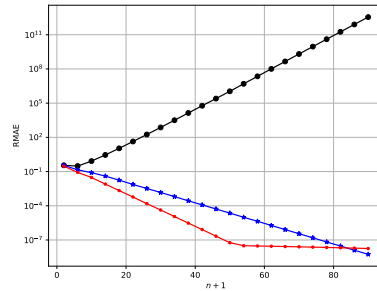For more experiments and details the reader can refer to [26].

Fig. 2 The RMAE for the function f_1 varying the number of nodes. The results with equispaced, CL and fake nodes are represented by black circles, blue stars and red dots, respectively.
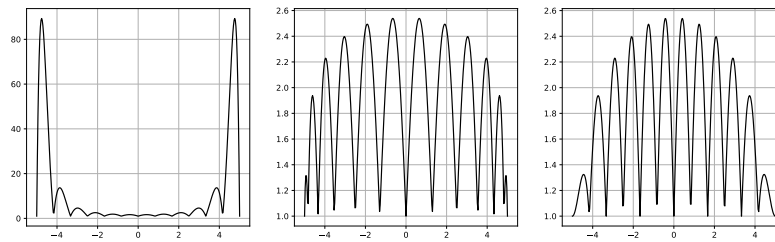

Fig. 3 Lebesgue functions of equispaced (left), CL (center) and fake CL (right) nodes.

3.1.2 Applications to Gibbs phenomenon

For the test, we consider the function with jump discontinuities at the origin

$$f_2(x) := \begin{cases} -\dfrac{x^2}{40} + 1, & x < 0, \\[2mm] \dfrac{x+4}{x+6}, & x \geq 0, \end{cases}$$

As done in the previous subsections 3.1.1, we compare the interpolant in the three different situations indicated in the gray box.

In Fig. 4 we display the results obtained using 20 interpolation points. We observe that the Gibbs phenomenon affects the reconstruction obtained via resampling on CL nodes, while it is well mitigated if using the fake nodes. In Fig. 5, we provide the RMAE behavior of these methods. The results are quite impressive, meaning that we are able to effectively reduce the Gibbs phenomenon by the S-Gibbs map of Algorithm 2.

Remarks. In the presence of discontinuities, it is interesting noticing the behaviour of the elementary Lagrange functions under the mapping approach.
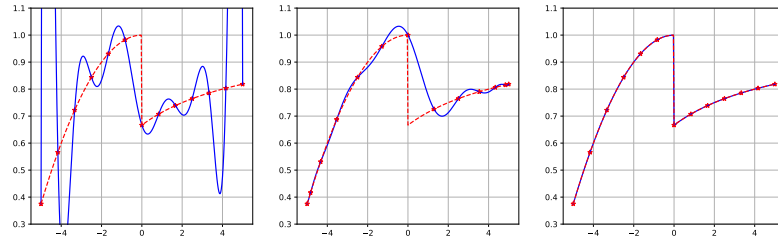
Fig. 4 Interpolation with 20 points of the function f_2 on $[-5, 5]$, using equispaced (left), CL nodes (center) and the discontinuous map (right). The nodes are represented by stars, the original and reconstructed functions are plotted with continuous red and dotted blue lines, respectively.
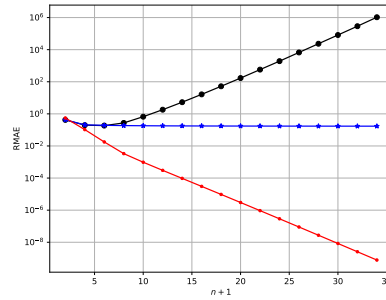

Fig. 5 The RMAE for the function f_2 varying the number of nodes. The results with equispaced, CL and fake nodes are represented by black circles, blue stars and red dots, respectively.

In Fig. 6 we show this effect for the cubic case when there is no jump and jump, with and without scaling. What is interesting to see is that the cardinal functions become discontinuous in the presence of a discontinuities or jumps. This is the idea that we will develop in Section 6, about the use of discontinuous kernels when treating discontinuous functions.

4 Application to rational approximation

We can extend the mapped bases or *"fake" nodes* approach to the barycentric rational interpolation. In particular we concentrate to the family of *Floater-Hormann (FH) rational approximants* [35] combined with *AAA-algorithm*. We consider the family of FH interpolants because they have shown good approximation properties for smooth functions, in particular using equidistant nodes (cf. [15]). The idea comes
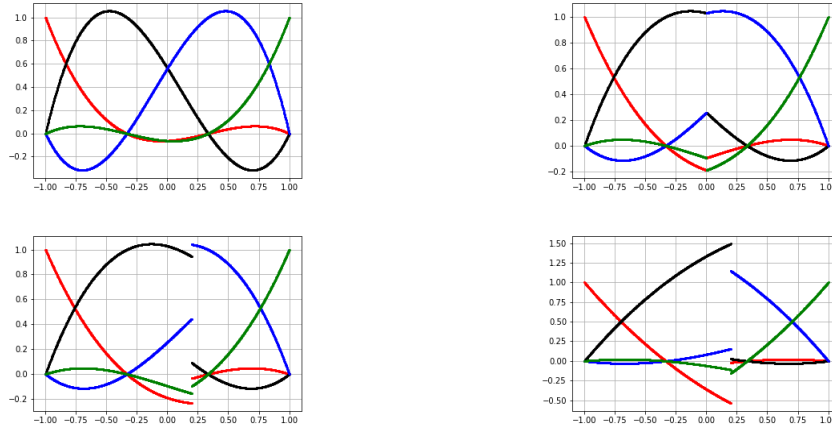
Fig. 6 Left-right, up-down: the original cardinals on 4 nodes, the cardinals around $\xi = 0$, $A = 1$ the cardinals around $\xi = 0.2$, $A = 1$, the cardinals around $\xi = 0.2$, $A = 5$.

from the fact that our previous construction is a general pourpose machinery, a kind of "black-box". Indeed, given a basis $\mathcal{B}$ for the approximation space, we can apply the S-map approach getting a new basis, say $\tilde{\mathcal{B}} = S \circ \mathcal{B}$. Considering that we are not resampling the values of the reconstructed function remains unchanged.

In [51], the *AAA-algorithm* has been discussed. This is a greedy algorithm to compute a barycentric rational approximant that is named by the authors *Adaptive Antoulas–Anderson algorithm*, reminding the names of the authors. This algorithm leads to impressively well-conditioned bases, which can be used in different fields, such as in computing conformal maps, or in rational minimax approximations (see e.g. [8]).

Unfortunately, both FH rational interpolants and those obtained by the AAA-algorithm, suffer of the *Gibbs phenomenon* when the underlying function presents jump discontinuities. For this reason we try to apply the *S-Gibbs* algorithm to mitigate or even better to reduce, this unpleasant effect.

We start by presenting a short overview of barycentric polynomial approximation and the FH family of rational approximants. Then we recall the main ideas about the AAA-algorithm and present the application of the S-Gibbs to both.

The numerical tests below will show an accurate interpolation of discontinuous functions.

4.1 Barycentric polynomial interpolation

As before let $X_n := \{x_i : i = 0, \ldots, n\}$ the set of $n+1$ distinct nodes in $I = [a,\ b] \subset \mathbb{R}$, increasingly ordered and let $f : I \longrightarrow \mathbb{R}$ be known at the sample points. Again we set $F_n := \{f_i = f(x_i) : i = 0, \ldots, n\}$.

It is well-known (see e.g. [10]) that it is possible to write the unique interpolating polynomial $P_{n;f}$ of degree at most n of f at X_n for any $x \in I$ in the second barycentric form

$$P_{n;f}(x) = \frac{\displaystyle\sum_{i=0}^{n} \frac{\lambda_i}{x - x_i} f_i}{\displaystyle\sum_{i=0}^{n} \frac{\lambda_i}{x - x_i}}, \tag{15}$$

where $\lambda_i = \displaystyle\prod_{j \neq i} \frac{1}{x_i - x_j}$ are called *weights* and this formula is one of the most stable way of evaluating $P_{n;f}$ (see [43]). If the weights λ_i are changed to other nonzero weights, say w_i, then the corresponding barycentric rational function

$$R_{n;f}(x) = \frac{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x - x_i} f_i}{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x - x_i}} \tag{16}$$

still satisfies the interpolation conditions $R_{n;f}(x_i) = f_i,\ i = 0, \ldots, n$.

4.2 Floater-Hormann rational interpolation

Let $n \in \mathbb{N}$, $d \in \{0, \ldots, n\}$. Let $p_i,\ i = 0, \ldots, n - d$ denote the unique polynomial interpolant of degree at most d interpolating the $d+1$ points $(x_k, f_k),\ k = i, \ldots, i+d$. The Floater-Hormann rational interpolant is

$$R_{n,d;f}(x) = \frac{\displaystyle\sum_{i=0}^{n-d} \lambda_i(x) p_i(x)}{\displaystyle\sum_{i=0}^{n-d} \lambda_i(x)}, \quad \text{where } \lambda_i(x) = \frac{(-1)^i}{(x - x_i) \cdots (x - x_{i+d})},$$

which interpolates f at the set of nodes X_n. It has been proved in [35] that $R_{n,d;f}$ has no real poles and that it reduces to the unique interpolating polynomial of degree at most n when $d = n$.

One can derive the barycentric form of this family of interpolants as well. Indeed, with considering the sets $J_i = \{k \in \{0, 1, \ldots, n - d\} : i - d \leq k \leq i\}$, one has

$$R_{n,d;f}(x) = \frac{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x - x_i} f_i}{\displaystyle\sum_{i=0}^{n} \frac{w_i}{x - x_i}}, \quad \text{where } w_i = (-1)^{i-d} \sum_{k \in J_i} \prod_{\substack{j=k \\ j \neq i}}^{j+d} \frac{1}{|x_i - x_j|}.$$

4.3 The AAA algorithm

Let us consider a set of points X_N with a large value of N (which represents the discretization of our domain) and a function $f : I \longrightarrow \mathbb{R}$. The *AAA-algorithm* (cf. [51]) is a *greedy* technique that at the step $m \geq 0$ considers the set $X^{(m)} = X_N \setminus \{x_0, \ldots, x_m\}$ and constructs the interpolant

$$R_{m;f}(x) = \frac{\displaystyle\sum_{i=0}^{m} \frac{w_i}{x - x_i} f_i}{\displaystyle\sum_{j=0}^{m} \frac{w_j}{x - x_j}} = \frac{n(x)}{d(x)},$$

by solving the discrete least squares problem for the weight vector $\boldsymbol{w} = (w_0, \ldots, w_m)$

$$\min \|fd - n\|_{X^{(m)}} \qquad \|\boldsymbol{w}\|_2 = 1,$$

where $\|\cdot\|_{X^{(m)}}$ is the discrete 2-norm over $X^{(m)}$. The next data site is $x_{m+1} \in X^{(m)}$ that makes the residual $|f(x) - n(x)/d(x)|$ maximum with respect to $x \in X^{(m)}$. This choice confirms the greedy nature of the process.

4.4 Mapped bases in barycentric rational interpolation

Using the ideas of mapped based approach in [26], we apply it to the *Floater-Hormann* interpolants and to its approximants produced via the *AAA algorithm*.

First, since the interpolant $R_{n;f}$ defined in (16) can be written using the cardinal basis, that is $R_{n;f}(x) = \sum_{j=0}^{n} f_j b_j(x)$, where

$$b_j(x) = \frac{\dfrac{w_j}{x - x_j}}{\sum_{i=0}^{n} \dfrac{w_i}{x - x_i}}.$$

is the j-th basis function, in the same spirit, we can write $R_{n;f}^S$ in the mapped cardinal basis form $R_{n;f}^S(x) = \sum_{i=0}^n f_i b_i^S(x)$, where $b_j^S(x) = \frac{\frac{w_j}{S(x)-S(x_j)}}{\sum_{i=0}^n \frac{w_i}{S(x)-S(x_i)}}$ is the j-th mapped basis function.

Using the S mapping approach, a more stable interpolant may arise. Indeed, the following result provides an upper bound for the S-Lebesgue constant that shows this improved stability.

Theorem 2 *Let* $\Lambda_n(X_n) = \max_{x \in I} \sum_{j=0}^n |b_j(x)|$ *and* $\Lambda_n^S(X_n) := \max_{x \in I} \sum_{j=0}^n |b_j^S(x)|$ *be the classical and the S-Lebesgue constants, respectively. We have*

$$\Lambda_n^S(X_n) \le C\Lambda_n(X_n),$$

where $C = \dfrac{\max_k M^k}{\min_k m_k}$ *with*

$$M^k = \max_{x \in I} \prod_{\substack{l=0 \\ l \ne k}}^n \left| \frac{S(x) - S(x_l)}{x - x_l} \right|, \quad m_k = \min_{x \in I} \prod_{\substack{l=0 \\ l \ne k}}^n \left| \frac{S(x) - S(x_l)}{x - x_l} \right|.$$

Proof As done in the polynomial case, we should bound each basis function b_j^S in terms of b_j for all $x \in I$. For details see [8].

Equivalently to the above mapped basis instance, we may construct the interpolant $R_{n;f}^S$ via the "fake" nodes approach just following the same ideas developed in Section 4.

Let $\tilde{R}_{n;g}$ be the barycentric rational interpolant as in (16) that interpolates, at the set of "fake "nodes $S(X_n)$, where the function $g : S(I) \longrightarrow \mathbb{R}$ interpolates the values F_n

$$g_{|S(X_n)} = f_{|X_n}.$$

Observe that $R_{n;f}^S(x) = \tilde{R}_{n;g}(S(x))$ for every $x \in I$. Hence, we may also build $R_{n;f}^S]$ upon a standard barycentric interpolation process, thereby providing a more intuitive interpretation of the method.

As we already observed, the choice of the mapping S is crucial for the accuracy interpolation process. Here, we confine our attention to the case in which f presents jump discontinuities and by using the *S-Gibbs* Algorithm (SGA), presented above, we construct an effective mapping S.

4.5 A numerical example

In $I = [-5, 5]$ we consider the discontinuous functions

$$f_1(x) = \begin{cases} e^{\frac{1}{x^2+1}}, & -5 \leq x < -3 \\ \sin(3x), & -3 \leq x < 2 \\ -\frac{x^3}{30} + 2, & 2 \leq x \leq 5. \end{cases} \qquad f_2(x) = \begin{cases} \cos(-\sin(x/2)), & -5 \leq x < -2.5 \\ \tan(x/2), & -2.5 \leq x < 2 \\ \log(e^{-\frac{1}{x-5.1}}), & 2 \leq x \leq 5. \end{cases}$$

We test the "fake" nodes approach with the S-Gibbs Algorithm in the framework of FH interpolants and the AAA algorithm for approximation. We fix $k = 10$ in the S-Gibbs. As observed in [26], also in this setting the choice of the shifting parameter is non-critical as long as it is taken "sufficiently large". We evaluate the constructed interpolants on a set of 5000 equispaced evaluation points $\Xi = \{\bar{x}_i = -5 + \frac{i}{1000} : i = 0, \ldots, 5000\}$ and compute the Relative Maximum Absolute Error RMAE as defined in (14, both for $R_{n;f}$ and $R_{n;f}^S$.

The FH interpolants

Here, we take various sets of equispaced nodes $X_n = \{-5 + \frac{5i}{n} : i = 0, \ldots, n\}$, varying the size of n. The results of the RMSE interpolation errors are displayed in Figures 4.5 and 4.5, by doubling the number of the nodes for $n = 40$ up to 2560. We simply observe that the proposed reconstruction via the fake nodes approach outperforms the standard technique.
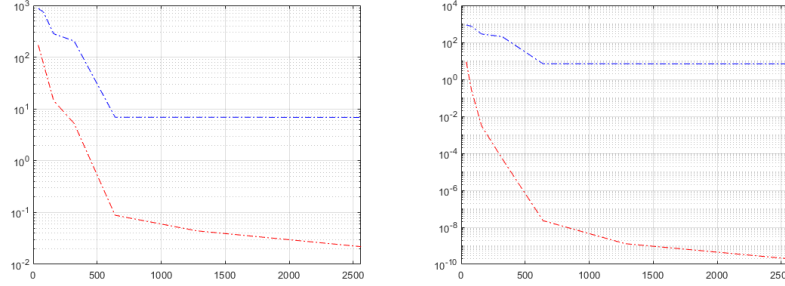

Fig. 7 RMSE interpolation error for f_1. Left with $d = 1$, right with $d = 4$. In blue, the standard interpolant $R_{n,d}$. In red, the proposed interpolant $R_{n,d}^s$.

The AAA algorithm

As the starting set for the AAA algorithm, we consider 10000 nodes randomly uniformly distributed in I, which we denote by X_{rand}.

Looking at Table 1, we observe that using the AAA algorithm with starting set $S(X_{rand})$ (indicated in the Table as AAA^S), that is, constructing the approximants via the fake nodes approach, does not suffer from the effects of the Gibbs phenomenon. For both approximants we fix the maximum degree to 20 and to 40 (by default 100 in the algorithm) getting rational approximants of type $(20, 20)$ and $(40, 40)$, respectively.
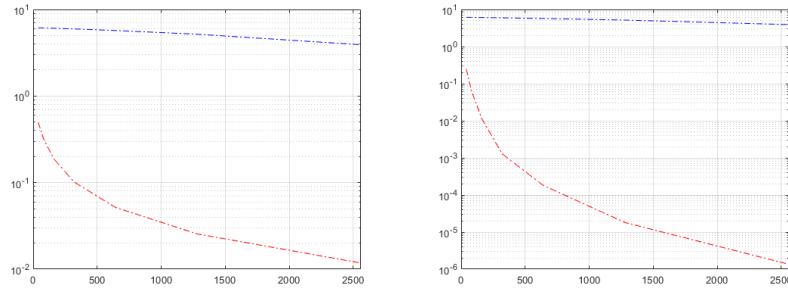
Fig. 8 RMSE interpolation error for f_2. Left with $d = 1$, right with $d = 4$. In blue, the standard interpolant $R_{n,d}$. In red, the proposed interpolant $R_{n,d}^s$.

f	m_{max}	AAA	AAA^s
f_1	20	5.8E+1	1.6E-4
	40	1.4E-1	2.5E-9
f_2	20	4.4E+1	7.4E-8
	40	8.0E-1	3.0E-12

Table 1 RMAE for AAA and AAA^s approximants

Remarks. This extension of the "fake" nodes approach to barycentric rational approximation, in particular to the family of FH interpolants, and its approximations by the AAA algorithm for the treatment of the Gibbs phenomenon via the S-Gibbs algorithm, shows that the proposed reconstructions outperform their classical versions, by erasing distortions and oscillations.

5 Application to quadrature

We now drive our attention towards the use of mapped bases as an alternative to standard quadrature rules as started to discuss in the ongoing paper [24].

Given $f : I = [a, b] \to \mathbb{R}$, we are interested in approximating

$$\mathcal{I}(f) := \int_I f(x)\,\mathrm{d}x,$$

via quadrature formulae of interpolation type. To this aim we take a set of distinct (quadrature) nodes $X_n = \{x_i, i = 0, \ldots, n\}$ by assuming $x_0 = a$, $x_n = b$ and $n \geq 1$.

The classical quadrature of interpolation type substitutes f by its interpolant $P_{n,f}(\cdot) = \sum_{i=0}^{n} a_i\, b_i(\cdot)$, with b_i the i-th basis element of the polynomial space $\mathbb{P}_n$ (note that the most simple basis one can consider is the monomials one, that is $b_i = x^k$) so that

$$\mathcal{I}(f) \approx \mathcal{I}_n(f) := \langle w, f \rangle. \tag{17}$$

The vector $\boldsymbol{f}$ contains the function values $f_k = f(x_k)$ while the vector of the weights $\boldsymbol{w} = (w_0, \ldots, w_n)^\intercal$ is computed by solving the moments' system

$$V^\intercal \, \boldsymbol{w} = \boldsymbol{m} \tag{18}$$

where V is the Vandermonde matrix so that $V_{i,j} = b_i(x_j)$, the vector $\boldsymbol{m} = (m_0, \ldots, m_n)^\intercal$ consists of the moments $m_k = \mathcal{I}(b_k), \quad k = 0, \ldots, n$.

Remark. The quadrature (17) is a scalar product that can be computed as the determinant of formula (5) substituting the elementary Lagrange polynomials with the weights and changing sing

$$\langle \boldsymbol{w}, \boldsymbol{f} \rangle = \det \begin{bmatrix} 0 & -w_0 & -w_1 & \cdots & -w_n \\ \hline f_0 & & & & \\ f_1 & & & I & \\ \vdots & & & & \\ f_n & & & & \end{bmatrix}. \tag{19}$$

5.1 Quadrature via "fake" nodes

In the mapped bases approach [26] instead of $P_{n;f}$ we consider the mapped interpolant $P_{n;f}^S$. It comes then "natural" to construct the quadrature formulae in the mapped basis, that is

$$\mathcal{I}(f) \approx \mathcal{I}_n^S(f) := \langle \boldsymbol{w}^S, \boldsymbol{f} \rangle .$$

The weights $\boldsymbol{w}^S = (w_0^S, \ldots, w_n^S)^\intercal$ are computed by solving the mapped-based system

$$(V^S)^\intercal \, \boldsymbol{w}^S = \boldsymbol{m}^S ,$$

with $\boldsymbol{m}^S$ the vector of S-mapped moments. For instance, taking the monomial basis, the i-th S-moment is $m_i^S = \mathcal{I}(S(x)^i)$.

Equivalently, for $x \in I$, the interpolant $P_{n;f}^S$ can be seen as a standard polynomial interpolant $P_{n;g}$ on the mapped set of nodes $S(I) = \hat{I}$. Moreover, if the map S is at least C^1 in I, letting $t = S(x)$, for $x \in I$, we get

$$\hat{S}(t) := \frac{\mathrm{d}S^{-1}(t)}{\mathrm{d}t} = \frac{1}{S'(S^{-1}(t))} . \tag{20}$$

Thus,

$$\mathcal{I}(P_{n;f}^S, I) = \mathcal{I}(P_{n;g}\hat{S}, \hat{I}) . \tag{21}$$

It then follows

Proposition 3 $w_i^S = \mathcal{I}(l_i \hat{S}, \hat{I})$.

Proof Considering the Lagrange basis of the polynomial interpolant so that

$$P_{n;g}(S(x)) = \sum_{i=0}^{n} f(x_i) l_i(S(x)) = \sum_{i=0}^{n} f(x_i) \prod_{\substack{0 \le j \le n \\ j \ne i}} \frac{S(x) - S(x_j)}{S(x_i) - S(x_j)},$$

and taken into account (20) we conclude   □

The following then can be easily proven.

Proposition 4 *Let S be such that g is at least* $C^{n+1}(\hat{I})$. *Then,*

$$E_n^S(f) = E_n(g) = \frac{g^{(n+1)}(S(\xi))}{(n+1)!} \mathcal{I}(\omega_{n+1}\hat{S}, \hat{I}), \quad \xi \in I$$

with $\omega_{n+1}(t) = \prod_{i=0}^{n}(t - t_i)$.

Proof Indeed

$$E_n^S(f) = \mathcal{I}((g - P_{n;g})\hat{S}, \hat{I}) = \frac{g^{(n+1)}(S(\xi))}{(n+1)!} \mathcal{I}(\omega_{n+1}\hat{S}, \tilde{I}),$$

as claimed.   □

Remark. We observe that if S is such that g is a polynomial of degree n, then the quadrature has exactness n.

We now focus on the computation of the weights for the two maps introduced above for S-Gibbs and S-Runge, respectively.

First, we recall that considering $h(x) = 1/\sqrt{1 - x^2}$, the Chebyshev weight function and C_n the CL nodes on $J = [-1, 1]$

$$\mathcal{I}(f\,h, J) \approx \sum_{i=0}^{n} f(c_i) w_i^c \tag{22}$$

where $w_i^c = \frac{\pi}{z_i n}$ with

$$z_i = \begin{cases} 2 & i = 0, n \\ 1 & \text{otherwise} \end{cases}$$

We are ready to prove the following.

Theorem 3 *Let X_n be the set of $n + 1$ equispaced points of $I = [a, b]$ and consider the S-Runge map*

$$S(x) = -\cos\left(\frac{x-a}{b-a}\pi\right). \tag{23}$$

Then,

$$w_k^S = \begin{cases} \frac{h}{2}, \textit{for } k \in \{0, n\}, \\ h, \textit{for } k = 1, \ldots, n-1. \end{cases}$$

Proof Let us now take on $I = [a, b]$ the set of equispaced nodes X_n and the S-map (23). Thus, $S(X_n) = C_n$ and letting $t = S(x)$ then $l_i(S(x)) = l_i(t)$, $i = 0, \ldots, n$, where $l_i(t)$ is the the Lagrange basis at the CL nodes of $J = [-1, 1]$. Therefore, by using (20) and (21)

$$dx = \hat{S}(t)\,dt = h(t)\frac{b-a}{\pi}\,dt,$$

we have

$$w_i^S = \frac{(b-a)}{\pi}\mathcal{I}(l_i\,h, J).$$

Finally, by observing that the quadrature rule (22) is exact for polynomials of degree n, we get

$$w_i^S = \frac{(b-a)}{\pi}\sum_{j=0}^{n} l_i(c_j)w_j^c = \frac{(b-a)}{\pi}\sum_{j=0}^{n}\delta_{i,j}\frac{\pi}{z_j n} = \frac{b-a}{z_i n}.$$

This concludes the proof. □

Remarks

1. This result turns out to be quite surprising but interesting: the weights of the "fake" nodes quadrature, coincide with those of the trapezoidal rule and up to the constant $\frac{\pi}{b-a}$ with those for the wighted quadrature based on the CL nodes in $[-1, 1]$.
2. In the case of composite quadrature rules, the weights for the "fake" nodes approach can be computed by applying the S-map on *each* subinterval.

5.1.1 Examples

We apply the "new" quadrature approach to the following test functions: a discontinuous one and a Runge-type one

$$f_1(x) = \begin{cases} \sin(x), & \text{for } x \le 0, \\ \log(x^4+4)+7, & \text{for } x > 0, \end{cases} \quad \text{and} \quad f_2(x) = \frac{1}{4x^2+1}.$$

We compute their integrals over the interval $I = [-2, 2]$. In Figure 9 the absolute error between the true value of the integral and its approximation is displayed. As

approximants we compare the standard approach on a equispaced points, the classical quadrature on Chebyshev-Lobatto nodes and the integral computed with the "fake" nodes approach: we use the S-Gibbs and S-Runge for f_1 and f_2, respectively. We consider polynomials with degrees $n = 2k$ with $k = 1, \ldots, 20$. We observe that the "fake" nodes quadrature outperforms the computation of the integral on equispaced nodes while still competitive with the classical Gaussian quadrature based on CL nodes.
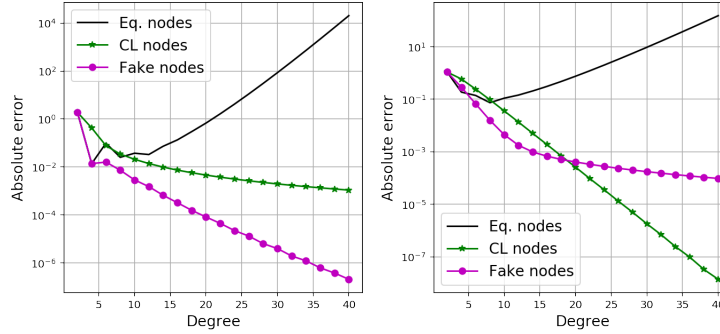

Fig. 9 Left: approximation of the integral over I of function f_1 and f_2, left and right respectively.

The experiments have been carried out in PYTHON 3.7 using NUMPY 1.15. The Python codes are available at https://github.com/pog87/FakeQuadrature.

6 Discontinuous kernels

Interpolation by kernels, mainly by radial kernels known as *Radial Basis Functions* (RBF), are suitable tools for high-dimensional scattered data problems, solution of PDES, machine learning, image registration. For an overview of the topic we refer the reader to the monographs [60, 34] and references therein. Our interest is now confined to the approximation of data with discontinuities. Indeed, based on recent studies on *Variably Scaled Kernels (VSKs)* [18, 55] and their discontinuous extension [28], we use discontinuous kernel functions that reflect discontinuities in the data as a basis. These basis functions, referred to as Variably Scaled Discontinuous Kernels (VSDKs), enable us naturally to interpolate functions with given discontinuities.

6.1 A brief introduction to RBF approximation

We start by introducing some basic notation and results about RBF interpolation.

Let $\mathcal{X}_N = \{x_i, \ i = 1, \ldots, N\}$ be a set of distinct data points (data sites or nodes) arbitrarily distributed on a domain $\Omega \subseteq \mathbb{R}^d$ and let $\mathcal{F}_N = \{f_i = f(x_i), \ i = 1, \ldots, N\}$ be an associated set of data values (measurements or function values) obtained by sampling some (unknown) function $f : \Omega \longrightarrow \mathbb{R}$ at the nodes x_i. We can reconstruct f by interpolation, that is by finding a function $\mathcal{P} : \Omega \longrightarrow \mathbb{R}$ satisfying the interpolation conditions

$$\mathcal{P}(x_i) = f_i, \quad i = 1, \ldots, N. \tag{24}$$

This problem (24) has unique solution provided $\mathcal{P} \in \mathrm{span}\{\Phi_\varepsilon(\cdot, x_i), x_i \in \mathcal{X}\}$, where $\Phi_\varepsilon : \Omega \times \Omega \longrightarrow \mathbb{R}$ is a strictly positive definite and symmetric kernel. Notice that Φ depends on the so-called *shape parameter* $\varepsilon > 0$ which allows to change the shape of Φ, making it flatter (or wider) as $\varepsilon \to 0+$ or spiky and so more localized as $\varepsilon \to +\infty$. This has important consequences in error analysis and stability of interpolation by RBF (cf. e.g. [58]).

The resulting kernel-based interpolant, denoted by $\mathcal{P}_{\varepsilon, \mathcal{X}_N}$, can be written as

$$\mathcal{P}_{\varepsilon, \mathcal{X}_N}(x) = \sum_{k=1}^N c_k \Phi_\varepsilon(x, x_k), \quad x \in \Omega. \tag{25}$$

The interpolation problem (24) in matrix form becomes $A_\varepsilon \in \mathbb{R}^{N \times N}$ with $(A_\varepsilon)_{ik} = \Phi_\varepsilon(x_i, x_k), \ i, k = 1, \ldots, N$. Then, letting $f = (f_1, \ldots, f_N)^\mathsf{T}$ the vector of data values, we can find the coefficients $c = (c_1, \ldots, c_N)^\mathsf{T}$ by solving the linear system $A_\varepsilon c = f$. Since we consider strictly positive definite and symmetric kernels, the existence and uniqueness of the solution of the linear system is ensured. More precisely, the class of strictly positive definite and symmetric *radial kernels* Φ_ε can be defined as follows.

Definition 2 Φ_ε is called radial kernel if there exists a continuous function $\varphi_\varepsilon : [0, +\infty) \longrightarrow \mathbb{R}$, depending on the shape parameter $\varepsilon > 0$, such that

$$\Phi_\varepsilon(x, y) = \varphi_\varepsilon(\|x - y\|_2), \tag{26}$$

for all $x, y \in \Omega$.

Remark. The notation (26) provides the "dimension-blindness" property of RBF. Hence, once we know the function φ and compose it with the Euclidean norm, we get a radial kernel.

In Table 2 we collect some of the most popular basis functions φ which are strictly positive definite. The Gaussian, Inverse Multiquadrics and Matérn (M0) are globally supported, while the Wendland (W2) and Buhmann (B2) are locally supported.

To Φ_ε we can associate a real pre-Hilbert space $H_{\Phi_\varepsilon}(\Omega)$

$$H_{\Phi_\varepsilon}(\Omega) = \mathrm{span}\{\Phi_\varepsilon(\cdot, x), \ x \in \Omega\},$$

Kernel	$\varphi(r)$
Gaussian C^∞ (GA)	$e^{-\varepsilon^2 r^2}$
Inverse Multiquadrics C^∞ (IM)	$(1 + r^2/\varepsilon^2)^{-1/2}$
Matérn C^0 (M0)	$e^{-\varepsilon r}$
Wendland C^2 (W2)	$(1 - \varepsilon r)_+^4 (4\varepsilon r + 1)$
Buhmann C^2 (B2)	$2r^4 \log r - 7/2r^4 + 16/3r^3 - 2r^2 + 1/6$

Table 2 Examples of strictly positive definite radial kernels depending on the shape parameter ε. The truncated power function is denoted by $(\cdot)_+$. W2 and B2 are compactly supported radial basis functions

with Φ_ε as reproducing kernel. This space will be then equipped with the bilinear form $(\cdot, \cdot)_{H_{\Phi_\varepsilon}(\Omega)}$. Then we define the associate *native space* $\mathcal{N}_{\Phi_\varepsilon}(\Omega)$ of Φ_ε as the completion of $H_{\Phi_\varepsilon}(\Omega)$ with respect to the norm $||\cdot||_{H_{\Phi_\varepsilon}(\Omega)}$, that is $||f||_{H_{\Phi_\varepsilon}(\Omega)} = ||f||_{\mathcal{N}_{\Phi_\varepsilon}(\Omega)}$ for all $f \in H_{\Phi_\varepsilon}(\Omega)$. For more details, as already quotes, see the monographs [34, 60]).

The accuracy of the interpolation process can be expressed, for instance, in terms of the *power function*. The power function is a positive function given as the ratio of two determinants (cf. [23])

$$P_{\Phi_\varepsilon, \mathcal{X}_N}(\boldsymbol{x}) := \sqrt{\frac{\det(A_\varepsilon(\mathcal{Y}_{N+1}))}{\det(A_\varepsilon(\mathcal{X}_N))}} \,. \qquad (27)$$

where $A_\varepsilon(\mathcal{X}_N)$ is the interpolation matrix related to the set of nodes $\mathcal{X}_N$ and the kernel Φ_ε and $A_\varepsilon(\mathcal{Y}_{N+1})$ the matrix associated to the set $\mathcal{Y}_{N+1} := \{\boldsymbol{x}\} \cup \mathcal{X}_N$, $\boldsymbol{x} \in \Omega$.

The following pointwise error bound holds.

Theorem 4 *Let $\Phi_\varepsilon \in C(\Omega \times \Omega)$ be a strictly positive definite kernel and $\mathcal{X}_N \subseteq \Omega$ a set of N distinct points. For all $f \in \mathcal{N}_{\Phi_\varepsilon}(\Omega)$ we have*

$$|f(\boldsymbol{x}) - \mathcal{P}_{\varepsilon, \mathcal{X}_N}(\boldsymbol{x})| \leq P_{\Phi_\varepsilon, \mathcal{X}_N}(\boldsymbol{x}) \, ||f||_{\mathcal{N}_{\Phi_\varepsilon}(\Omega)}, \quad \boldsymbol{x} \in \Omega.$$

Remarks. We observe that this Theorem bounds the error in terms of the power function which depends on the kernel and the data points but is independent on the function values.

This theorem is a special instance of [60, Theorem 11.3, p. 182] where the fill-distance

$$h_{\mathcal{X}_N, \Omega} := \sup_{\boldsymbol{x} \in \Omega} \min_{\boldsymbol{x}_k \in \mathcal{X}_N} ||\boldsymbol{x} - \boldsymbol{x}_k||,$$

is used instead of the power function.

6.2 From RBF to VSK interpolation

The choice of the shape parameter ε is a crucial computational issue in RBF inter-
polation leading to instability effects without a clever choice of it. This is usually
done by analyzing the behaviour of some kind of errors (like the Root Mean Square
Error) versus the conditioning of the interpolation matrix (cf. e. g. [38, 52]) and so
many techniques has been developed in order to overcome such problems. Many of
these techniques allow to choose the *best* shape parameter based on a "trade-off"
between conditioning and efficiency. There are approaches based on the choice of
well-conditioned bases, like in the RBF-QR method for Gaussians [37] or in the
more general setting discussed in [30].

In the seminal paper[18] the authors introduced the so called *Variably Scaled
Kernels (or VSKs)* where the classical tuning strategy of finding the optimal shape
parameter, is substituted by the choice of a *scale function* which plays the role of a
density function. More precisely [18, Def. 2.1]

Definition 3 Letting $I \subseteq (0, +\infty)$ and Φ_ε a positive definite radial kernel on $\Omega \times I$
depending on the shape parameter $\varepsilon > 0$. Given a scale function $\psi : \Omega \longrightarrow I$, a
Variably Scaled Kernel Φ_ψ on Ω is

$$\Phi_\psi(x, y) := \Phi_1((x, \psi(x)), (y, \psi(y))), \tag{28}$$

for $x, y \in \Omega$.

Defining then the map $\Psi(x) = (x, \psi(x))$ on Ω, the interpolant on the set of nodes
$\Psi(X_N) := \{(x_k, \psi(x_k)), \ x_k \in X_N\}$ with fixed shape parameter $\varepsilon = 1$ (or any other
constant value c) takes the form

$$\mathcal{P}_{1,\Psi(X_N)}(\Psi(x)) = \sum_{k=1}^{N} c_k \Phi_1(\Psi(x), \Psi(x_k)), \tag{29}$$

with $x \in \Omega, \ x_k \in X_N$.

By analogy with the interpolant in (25), the vector of coefficients $c = (c_1, \ldots, c_N)^\top$
is determined by solving the linear system $A_\psi c = f$, where f is the vector of data
values and $(A_\psi)_{ik} = \Phi_\psi(x_i, x_k)$ is strictly positive definite because of the strictly
positiveness of Φ_ψ.

Once we have the interpolant $\mathcal{P}_{1,\Psi(X_N)}$ on $\Omega \times I$, we can project back on Ω the
points $(x, \psi(x)) \in \Omega \times I$. In this way, we obtain the so-called VSK interpolant $\mathcal{V}_\psi$
on Ω. Indeed, by using (28), we get

$$\mathcal{V}_\psi(x) := \sum_{k=1}^{N} c_k \Phi_\psi(x, x_k) = \sum_{k=1}^{N} c_k \Phi_1(\Psi(x), \Psi(x_k)) = \mathcal{P}_{1,\Psi(X_N)}(\Psi(x)). \tag{30}$$

The error and stability analysis of this varying scale process on Ω coincides with the
analysis of a fixed scale kernel on $\Omega \times I$ (for details and analysis of these continuous
scale functions see [18]).

6.3 Variably Scaled Discontinuous Kernels (VSDK)

To understand the construction of a VSDK let start from the one dimensional case. Let $\Omega = (a, b) \subset \mathbb{R}$ be an open interval, $\xi \in \Omega$ and the discontinuous function $f : \Omega \longrightarrow \mathbb{R}$

$$f(x) := \begin{cases} f_1(x), & a < x < \xi, \\ f_2(x), & \xi \le x < b, \end{cases}$$

where f_1, f_2 are real valued smooth functions for which exist finite the limits $\lim_{x \to a^+} f_1(x)$, $\lim_{x \to b^-} f_2(x)$ and $f_2(\xi) \ne \lim_{x \to \xi} f_1(x)$.

If we approximate f on some set of nodes, say $\mathcal{X} \subset \Omega$, the presence of a jump will result in unpleasant oscillations due to the *Gibbs phenomenon*. The idea is then to approximate f at $\mathcal{X}$ by interpolants of the form (30) with the main issue of considering discontinuous scale functions in the interpolation process. The strategy is that of approximating a function having jumps by using a scale function that has jumps discontinuities at the same positions of the considered function.

To this aim, take $\alpha, \beta \in \mathbb{R}$, $\alpha \ne \beta$, $\mathcal{S} = \{\alpha, \beta\}$ and the scale function $\psi : \Omega \longrightarrow \mathcal{S}$

$$\psi(x) := \begin{cases} \alpha, & x < \xi, \\ \beta, & x \ge \xi. \end{cases}$$

which is piecewise constant, having a discontinuity exactly at ξ as the function f.

Then we consider Φ_ε a positive definite radial kernel on $\Omega \times \mathcal{S}$, possibly depending on a shape parameter $\varepsilon > 0$ or alternatively a VSK Φ_ψ on Ω as in (28). For the sake of simplicity we start by taking the function φ_1 related to the kernel Φ_1 that is

$$\varphi_1(\|\Psi(x) - \Psi(y)\|_2) = \varphi_1(\|(x, \psi(x)) - (y, \psi(y))\|_2) =$$
$$= \varphi_1 \left(\sqrt{(x - y)^2 + (\psi(x) - \psi(y))^2} \right),$$

so that

$$\varphi_1(\|\Psi(x) - \Psi(y)\|_2) = \begin{cases} \varphi_1(|x - y|), & x, y < \xi \quad \text{or} \quad x, y \ge \xi, \\ \varphi_1(\|(x, \alpha) - (y, \beta)\|_2), & x < \xi \le y \quad \text{or} \quad y < \xi \le x, \end{cases}$$

noticing that $\varphi_1(\|(x, \alpha) - (y, \beta)\|_2) = \varphi_1(\|(x, \beta) - (y, \alpha)\|_2)$.

Then, consider the set $\mathcal{X}_N = \{x_k, \ k = 1, \dots, N\}$ of points of Ω and the interpolant $\mathcal{V}_\psi : \Omega \longrightarrow \mathbb{R}$ which is a linear combination of discontinuous functions $\Phi_\psi(\cdot, x_k)$ having a jump at ξ.

- if $a < x_k < \xi$

$$\Phi_\psi(x, x_k) = \begin{cases} \varphi_1(|x - x_k|), & x < \xi, \\ \varphi_1(\|(x, \alpha) - (x_k, \beta)\|_2), & x \ge \xi, \end{cases}$$

- if $\xi \le x_k < b$

$$\Phi_\psi(x, x_k) = \begin{cases} \varphi_1(|x - x_k|), & x \geq \xi, \\ \varphi_1(\|(x, \alpha) - (x_k, \beta)\|_2), & x < \xi. \end{cases}$$

By this construction we can give the following definition that extends the idea when we have more than one jump.

Definition 4 Let $\Omega = (a, b) \subset \mathbb{R}$ be an open interval, $\mathcal{S} = \{\alpha, \beta\}$ with $\alpha, \beta \in \mathbb{R}_{>0}$, $\alpha \neq \beta$ and let $\mathcal{D} = \{\xi_j, \ j = 1, \ldots, \ell\} \subset \Omega$ be a set of distinct points with $\xi_j < \xi_{j+1}$ for every j. Let $\psi : \Omega \longrightarrow \mathcal{S}$ the scale function defined as

$$\psi(x) := \begin{cases} \alpha, & x \in (a, \xi_1) \text{ or } x \in [\xi_j, \xi_{j+1}), & \text{where } j \text{ is even}, \\ \beta, & x \in [\xi_j, \xi_{j+1}), & \text{where } j \text{ is odd}, \end{cases}$$

and

$$\psi(x)_{|[\xi_\ell, b)} := \begin{cases} \alpha, & \ell \text{ is even}, \\ \beta, & \ell \text{ is odd}. \end{cases}$$

The kernel Φ_ψ is then called a Variably Scaled Discontinuous Kernel on Ω.

For the analysis of the VSDKs introduced in Definition 4 we can not rely on some important and well-known results of RBF interpolation due to the discontinuous nature of these kernels. Therefore we may proceed as follows.

Let Ω and $\mathcal{D}$ be as in Definition 4 and $n \in \mathbb{N}$. We define $\psi_n : \Omega \longrightarrow \mathcal{I} \subseteq (0, +\infty)$ as

$$\psi_n(x) := \begin{cases} \alpha, & x \in (a, \xi_1 - 1/n) \text{ or } x \in [\xi_j + 1/n, \xi_{j+1} - 1/n) & j \text{ is even}, \\ \beta, & x \in [\xi_j + 1/n, \xi_{j+1} - 1/n) & j \text{ is odd}, \\ \gamma_1(x), & x \in [\xi_j - 1/n, \xi_j + 1/n) & j \text{ is odd}, \\ \gamma_2(x), & x \in [\xi_j - 1/n, \xi_j + 1/n) & j \text{ is even}, \end{cases}$$
$$(31)$$
$$\psi_n(x)_{|[\xi_\ell + 1/n, b)} := \begin{cases} \alpha, & \ell \text{ is even}, \\ \beta, & \ell \text{ is odd}, \end{cases}$$

where γ_1, γ_2 are continuous, strictly monotonic functions so that

$$\lim_{x \to \xi_{j+1} + 1/n} \gamma_1(x) = \gamma_2(\xi_j - 1/n) = \beta, \qquad \lim_{x \to \xi_{j+1} + 1/n} \gamma_2(x) = \gamma_1(\xi_j - 1/n) = \alpha.$$

From Definition 4, it is straightforward to verify that $\forall x \in \Omega$ the following pointwise convergence result holds

$$\lim_{n \to \infty} \psi_n(x) = \psi(x).$$

We point out that for every fixed $n \in \mathbb{N}$ the kernel Φ_{ψ_n} is a continuous VSK, hence it satisfies the error bound of Theorem 4. For VSDKs instead we have the following results whose proofs can be found in the paper [28].

Theorem 5 *For every $x, y \in \Omega$, we have*

$$\lim_{n \to \infty} \Phi_{\psi_n}(x, y) = \Phi_\psi(x, y),$$

where Φ_ψ is the kernel considered in Definition 4.

Corollary 1 *Let* $H_{\Phi_{\psi_n}}(\Omega) = span\{\Phi_{\psi_n}(\cdot, x), \ x \in \Omega\}$ *be equipped with the bilinear form* $(\cdot, \cdot)_{H_{\Phi_{\psi_n}}(\Omega)}$ *and let* $N_{\Phi_{\psi_n}}(\Omega)$ *be the related native space. Then, taking the limit of the basis functions, we obtain the space* $H_{\Phi_\psi}(\Omega) = span\{\Phi_\psi(\cdot, x), \ x \in \Omega\}$ *equipped with the bilinear form* $(\cdot, \cdot)_{H_{\Phi_\psi}(\Omega)}$ *and the related native space* $N_{\Phi_\psi}(\Omega)$.

We get an immediate consequence for the interpolant $\mathcal{V}_\psi$ too.

Corollary 2 *Let* Ω, $\mathcal{S}$ *and* $\mathcal{D}$ *be as in Definition 4. Let* $f : \Omega \longrightarrow \mathbb{R}$ *be a discontinuous function whose step discontinuities are located at the points belonging to* $\mathcal{D}$. *Moreover, let* ψ_n *and* ψ *be as in Theorem 5. Then, considering the interpolation problem with nodes* $\mathcal{X}_N = \{x_k, \ k = 1, \ldots, N\}$ *on* Ω, *we have*

$$\lim_{n \to \infty} \mathcal{V}_{\psi_n}(x) = \mathcal{V}_\psi(x),$$

for every $x \in \Omega$.

To provide error bounds in terms of the *power function*, we should first define the equivalent power function for a VSDK Φ_ψ on a set of nodes $\mathcal{X}_N$. From (27), we immediately have

$$P_{\Phi_\psi, \mathcal{X}}(x) = \sqrt{\frac{\det(A_\psi(\mathcal{Y}_{N+1}))}{\det(A_\psi(\mathcal{X}_N))}}.$$

From Theorem 5 and Corollary 1, we may define the power function for a discontinuous kernel as

$$P_{\Phi_\psi, \mathcal{X}_N}(x) = \lim_{n \to \infty} P_{\Phi_{\psi_n}, \mathcal{X}_N}(x), \ \forall x \in \Omega.$$

Finally, we can state the error bound for interpolation via VSDKs.

Theorem 6 *Let* Φ_ψ *be a VSDK on* $\Omega = (a, b) \subset \mathbb{R}$, *and* $\mathcal{X}_N \subseteq \Omega$ *consisting of N distinct points. For all* $f \in N_{\Phi_\psi}(\Omega)$ *we have*

$$|f(x) - \mathcal{V}_\psi(x)| \le P_{\Phi_\psi, \mathcal{X}_N}(x)\|f\|_{N_{\Phi_\psi}(\Omega)}, \quad x \in \Omega.$$

Proof For all $n \in \mathbb{N}$ and $x \in \Omega$, since the VSK Φ_{ψ_n} is continuous, from Theorem 4, we get

$$|f(x) - \mathcal{V}_{\psi_n}(x)| \le P_{\Phi_{\psi_n}, \mathcal{X}}(x)\|f\|_{N_{\Phi_{\psi_n}}(\Omega)}.$$

Then, taking the limit $n \to \infty$ and recalling the results of this subsection, the thesis follows. $\square$

6.4 VSDKs: multidimensional case

VSDKs rely upon the classical RBF bases, therefore as noticed are "dimension-blind" which make them a suitable and flexible tool to approximate data in any dimension.

However, in higher dimensions, the geometry is more complex, so we must pay attention in defining the scale function ψ on a bounded domain $\Omega \subset \mathbb{R}^d$.

We consider the following setting.

(i) The bounded set $\Omega \subset \mathbb{R}^d$ is the union of n pairwise disjoint sets Ω_i and $\mathcal{P}_\Omega = \{\Omega_1, \ldots, \Omega_n\}$ forms a partition of Ω.

(ii) The subsets Ω_i satisfy an interior cone condition and have a Lipschitz boundary.

(iii) Let $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ and $\Sigma = \{\alpha_1, \ldots, \alpha_n\}$. The function $\psi : \Omega \to \Sigma$ is constant on the subsets Ω_i, i.e., $\psi(\boldsymbol{x}) = \alpha_i$ for all $\boldsymbol{x} \in \Omega_i$. In particular, ψ is piecewise constant on Ω and discontinuities appear only at the boundaries of the subsets Ω_i. We further assume that $\alpha_i \neq \alpha_j$ if Ω_i and Ω_j are neighboring sets.

A suitable scale function ψ for interpolating f via VSDKs on $\Omega \subset \mathbb{R}^d$ can be defined as follows.

Let $\Omega \subset \mathbb{R}^d$ satisfies the assumptions (i)–(iii) above. We define the scale function $\psi : \Omega \longrightarrow \mathcal{S}$ as

$$\psi_{|\Omega_i} := \alpha_i. \tag{32}$$

Definition 5 Given the scale function (32) the kernel Φ_ψ defined by (28) is a Variably Scaled Discontinuous Kernel on Ω.

Remarks. In Definition 5 we choose a scale function which emulates the properties of the one-dimensional function of Definition 4. The difference is that the multidimensional ψ could be discontinuous not just at the same points as f, but also on other nodes. That is all the jumps of f lye along $(d-1)$-dimensional manifolds $\gamma_1, \ldots, \gamma_p$ which verify

$$\gamma_i \subseteq \bigcup_{i=1}^{n} \partial\Omega_i \setminus \partial\Omega, \quad \forall i = 1, \ldots, p.$$

More precisely, if we are able to choose $\mathcal{P}_\Omega$ so that

$$\bigcup_{i=1}^{p} \gamma_i = \bigcup_{i=1}^{n} \partial\Omega_i \setminus \partial\Omega,$$

then f and ψ have the same discontinuities. Otherwise, if

$$\bigcup_{i=1}^{p} \gamma_i \subset \bigcup_{i=1}^{n} \partial\Omega_i \setminus \partial\Omega,$$

then ψ is discontinuous along $\bigcup_{i=1}^{n} \partial\Omega_i \setminus \left(\partial\Omega \cup \bigcup_{i=1}^{p} \gamma_i\right)$, while f is not.

The theoretical analysis in the multidimensional case is done along the same path of the one-dimensional setting. The idea is to consider continuous scale functions $\psi_n : \Omega \longrightarrow \mathcal{I} \subseteq (0, +\infty)$ such that $\forall \boldsymbol{x} \in \Omega$ the limits hold

$$\lim_{n\to\infty} \psi_n(\boldsymbol{x}) = \psi(\boldsymbol{x}),$$

and

$$\lim_{n\to\infty} \mathcal{V}_{\psi_n}(\boldsymbol{x}) = \mathcal{V}_{\psi}(\boldsymbol{x}).$$

We omit this extension and all the corresponding considerations which are similar to those discussed above for the one-dimensional setting, while we state directly the error bound.

Theorem 7 *Let Φ_ψ be a VSDK as in Definition 5. Suppose that $X_N = \{\boldsymbol{x}_i, i = 1, \dots, N\} \subseteq \Omega$ have distinct points. For all $f \in \mathcal{N}_{\Phi_\psi}(\Omega)$ we have*

$$|f(\boldsymbol{x}) - \mathcal{V}_\psi(\boldsymbol{x})| \le P_{\Phi_\psi, X_N}(\boldsymbol{x}) \|f\|_{\mathcal{N}_{\Phi_\psi}(\Omega)}, \quad \boldsymbol{x} \in \Omega.$$

Proof Just refer to Theorem 6 and considerations made above.   □

For the characterization of the native spaces for the VSDKs (if the discontinuities are known) and Sobolev-type error estimates, based on the fill-distance, of the respective interpolation scheme the reader must refer to the very recent paper [27].

7 Application to MPI

As we already observed, interpolation is an essential tool in medical imaging. It is required for example in geometric alignment or registration of images, to improve the quality of images on display devices, or to reconstruct the image from a compressed amount of data. In Computerized Tomography (CT) and Magnetic Resonance Imaging (MRI), which are examples of medical inverse problems, interpolation is used in the reconstruction step in order to fit the discrete Radon data into the back projection step. Similarly in SPECT for regridding the projection data in order to improve the reconstruction quality while reducing the acquisition computational cost [59]. In Magnetic Particle Imaging (MPI), the number of calibration measurements can be reduced by using interpolation methods, as well (see the important paper [46]).

In the early 2000s, B. Gleich and J. Weizenecker [39], invented at Philips Research in Hamburg a new quantitative imaging method called Magnetic Particle Imaging (MPI). In this imaging technology, a tracer consisting of superparamagnetic iron oxide nanoparticles is injected and then detected through the superimposition of different magnetic fields. In common MPI scanners, the acquisition of the signal is performed following a generated field free point (FFP) along a chosen sampling trajectory. The determination of the particle distribution given the measured voltages in the receive coils is an ill-posed inverse problem that can be solved only with proper regularization techniques [47].

Commonly used trajectories in MPI are *Lissajous curves* [48], which are parametric space-filling curves of the square $Q_2 = [-1, 1]^2$. More precisely, by using the same notations in [31, 32], let $\boldsymbol{n} = (n_1, n_2) \in \mathbb{N}^2$ be a vector of relatively prime

integers and $\epsilon \in \{1, 2\}$, the two-dimensional Lissajous curve $\gamma_\epsilon^n : [0, 2\pi] \to Q_2$ is defined as

$$\gamma_\epsilon^n(t) := (\cos(n_2 t), \cos(n_1 t - (\epsilon - 1)\pi/(2n_2))) .$$

The curve γ_ϵ^n is called *degenerate* if $\epsilon = 1$, and *non-degenerate* if $\epsilon = 2$. The *Padua points* of degree n are a degenerate Lissajous curve which have generating curve $\gamma_{Pad}(t) = (-\cos((n + 1)t), -\cos(nt)), \quad t \in [0, \pi]$ (see also [11, 14]).

The set of Lissajous node points associated to the curve γ_ϵ^n is given by

$$\mathrm{LS}_\epsilon^n := \left\{ \gamma_\epsilon^n\left(\tfrac{\pi k}{\epsilon n_1 n_2}\right) : \qquad k = 0, \ldots, 2\epsilon n_1 n_2 - 1 \right\} . \tag{33}$$

We define also the index set $\Gamma^{2n} := \left\{ (i, j) \in \mathbb{N}_0^2 : (i/2n_1) + (j/2n_2) < 1 \right\} \cup \{(0, 2n_2)\}$

which give the cardinality of the set, that is

$$\#\mathrm{LS}_\epsilon^n = \frac{(\epsilon n_1 + 1)(\epsilon n_2 + 1) - (\epsilon - 1)}{2} .$$
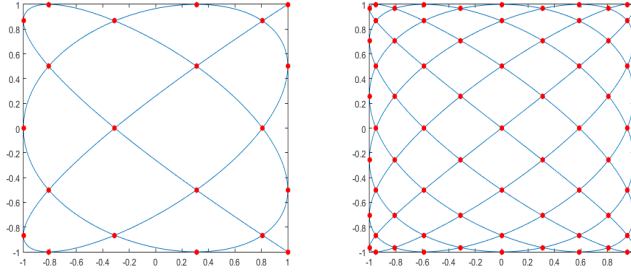

Fig. 10 Left: The degenerate curve $\gamma_1^{(5,6)}$. Right: the non-degenerate curve $\gamma_2^{(5,6)}$

To reduce the amount of calibration measurements, it is shown in [46] that the reconstruction can be restricted to particular sampling points along the Lissajous curves, i.e., the Lissajous nodes $\mathbf{LS}_2^{(n)}$ introduced in (33). Furthermore, by using a polynomial interpolation method on the Lissajous nodes the entire density of the magnetic particles can then be restored (cf. [31]). As noticed, these sampling nodes and the corresponding polynomial interpolation are an extension of the theory of the *Padua points* (see [11, 14] and also https://en.wikipedia.org/wiki/Padua_points).

If the original particle density has sharp edges, the polynomial reconstruction scheme on the Lissajous nodes is affected by the Gibbs phenomenon. As shown in [25], post-processing filters can be used to reduce oscillations for polynomial reconstruction in MPI. In the following, we demonstrate that the usage of the VSDK interpolation method in combination with the presented edge estimator effectively avoids ringing artifacts in MPI and provides reconstructions with sharpened edges.

7.1 An example

As a test data set, we consider MPI measurements conducted in [46] on a phantom consisting of three tubes filled with Resovist, a contrast agent consisting of superparamagnetic iron oxide. By the proceeding described in [46] we then obtain a reconstruction of the particle density on the Lissajous nodes $\mathbf{LS}_2^{(33,32)}$ (due to the scanner available, as described in [46]). This reduced reconstruction on the Lissajous nodes is illustrated in Figure 11 (left). A computed polynomial interpolant of this data is shown in Figure 11 (middle, left). In this polynomial interpolant some ringing artifacts are visible The scaling function ψ for the VSDK scheme is then obtained by using the classification Algorithm [27] with a Gaussian function for the kernel machine. The resulting scaling function is visualized in Figure 11 (middle, right). Using the C^0-Matérn (M0) kernel (see Table 2) for the VSDK interpolation, the final interpolant for the given MPI data is shown in in Figure 11 (right).

Fig. 11 Comparison of different interpolation methods in MPI. The reconstructed data on the Lissajous nodes $\mathbf{LS}_2^{(33,32)}$ (left) is first interpolated using the polynomial scheme derived in [31] (middle left). Using a mask constructed upon a threshold strategy described in [27] (middle right) the second interpolation is performed by the VSDK scheme (right).

8 Conclusion and further works

We have investigated the application of the polynomial mapped bases approach without resampling for reducing the Runge and Gibbs phenomena. The approach shows to be a kind of black-box that can be applied in many other frameworks. We indeed have applied it to barycentric rational approximation and quadrature. We have also studied the use of VSDK, a new family of variable scaled kernels, particularly effective in the presence of discontinuity in our data. A particular applications of VSDK is the image reconstruction from data coming from MPI scanners acquisitions.

Concerning the work in progress and the future works

- In the 2d case, we have results on discontinuous functions on the square, using polynomial approximation at the Padua points or tensor product meshes. In Figure 12 we show the results of the interpolation of a discontinuous function along a disk of the square $[-1,1]^2$, where the reconstruction has been done

by interpolation on the Padua points of degree 60 on the left. On the right we show the same reconstruction where the points that do not fall inside the disk are mapped with a circular mapping. The mapping strategy indeed reduce the Gibbs oscillations, but outside the disk we can not interpolate bu we can approximate by least-squares, because of the "fake Padua" points that are not anymore unisolvent.
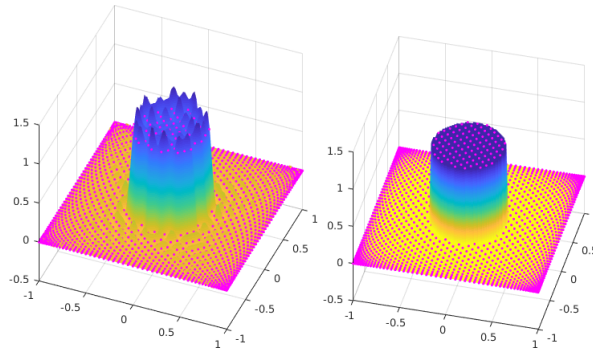

Fig. 12 Left: interpolation with Padua points of degree 60 of a function with a circular jump. Right: the same by mapping circularly the PD points, and using least-squares "fake-Padua"

- Again in 2d but also in 3d we can extract the so called *Approximate Fekete Points* of *Discrete Leja sequences* (cf. [13]) on various domains (disk, sphere, polygons, spherical caps, lunes and other domains). These points are numerically computed by numerical linear algebra methods and extracted from the so called *weakly admissible meshes* (WAM). For details about WAMs, refer to fundamental paper [12]

Finally we are working in improving the error analysis and finding more precise bounds for the Lebesgue constant(s). Among the applications of this approach we are interested to image registration in nuclear medicine and the reconstruction of periodic signals.

References

1. B. Adcock, R.B. Platte: *A mapped polynomial method for high-accuracy approximations on arbitrary grids*, SIAM J. Numer. Anal. **54** (2016), 2256–2281.
2. S. Amat, J. Ruiz, J.C. Trillo, D.F. Yáñez: *Analysis of the Gibbs phenomenon in stationary subdivision schemes*, Appl. Math. Lett. **76** (2018), 157–163.
3. S. Amat, K. Dadourian, J. Liandrat: *On a nonlinear subdivision scheme avoiding Gibbs oscillations and converging towards C^s functions with $s > 1$*, Math. Comput. **80** (2011), 959–971.
4. R. Archibald, A. Gelb, J. Yoon: *Fitting for edge detection in irregularly sampled signals and images*, SIAM J. Numer. Anal. **43** (2005), 259–279.
5. A. Bayliss, E. Turkel: *Mappings and accuracy for Chebyshev pseudo-spectral approximations*, J. Comput. Phys. **101** (1992), 349–359.
6. Bernhard Beckermann, Ana C. Matos and Franck Wielonsky, *Reduction of the Gibbs phenomenon for smooth functions with jumps by the ϵ-algorithm*, J. Comput. Appl. Math. 219(2), (2008), 329–349.
7. J.P. Berrut, G. Elefante: *A periodic map for linear barycentric rational trigonometric interpolation*, Appl. Math. Comput. 371 (2020) https://doi.org/10.1016/j.amc.2019.124924.
8. J.P. Berrut, S. De Marchi, G. Elefante and F. Marchetti: *Treating the Gibbs phenomenon in barycentric rational interpolation via the S-Gibbs algorithm*, Appl. Math. Letters 103 (2020) 106196.
9. J. P. Berrut and H. D. Mittelmann: *Lebesgue constant minimizing linear rational interpolation of continuous functions over the interval*, Comput. Math. Appl. **33** (1997), 77–86.
10. J. P. Berrut and L. N. Trefethen: *Barycentric Lagrange interpolation*, SIAM Rev. **46** (2004), 501–517.
11. L. Bos, M. Caliari, S. De Marchi, M. Vianello, Y. Xu, *Bivariate Lagrange interpolation at the Padua points: the generating curve approach*, J. Approx. Theory 143 (2006), 15–25.
12. L. Bos, J.-P. Calvi, N. Levenberg, A. Sommariva and M. Vianello, Geometric Weakly Admissible Meshes, Discrete Least Squares Approximation and Approximate Fekete Points, Math. Comp. 80 (2011), 1601–1621.
13. L. Bos, S. De Marchi, A. Sommariva and M. Vianello, Computing multivariate Fekete and Leja points by numerical linear algebra, SIAM J. Numer. Anal. 48 (2010), 1984–1999.
14. L. Bos, S. De Marchi, M. Vianello: *Polynomial approximation on Lissajous curves in the $d-$cube*. Appl. Numer. Math. 116 (2017), 47–56.
15. L. Bos, D. De Marchi, K. Hormann: *On the Lebesgue constant of Berrut's rational interpolant at equidistant nodes*, J. Comput. Appl. Math. **236** (2011), 504–510.
16. L. Bos, S. De Marchi, K. Hormann and J. Sidon: *Bounding the Lebesgue constant of Berrut's rational interpolant at general nodes*, J. Approx. Theory Vol. 169 (2013), 7–22
17. J.P. Boyd, F. Xu, *Divergence (Runge Phenomenon) for least-squares polynomial approximation on an equispaced grid and Mock-Chebyshev subset interpolation*, Appl. Math. and Comput. 210 (2009), 158–168.
18. M. Bozzini, L. Lenarduzzi, M. Rossini, R. Schaback: *Interpolation with variably scaled kernels*, IMA J. Numer. Anal.35(2015), 199–219.
19. L. Brutman: *Lebesgue functions for polynomial interpolation - a survey*, Annals of Numerical Mathematics, 4 (1997), 111–127.
20. E. W. Cheney: Introduction to Approximation Theory, AMS Chelsea Publishing, Second Edition, reprinted 1998.
21. P. J. Davis: Interpolation and Approximation, Blaisdell, New York, 1963.
22. S. De Marchi: *Polynomials arising in factoring generalized Vandermonde determinants: an algorithm for computing their coefficients*, Math. Comput. Modelling **34** (2001), 271–281.
23. S. De Marchi: *On optimal center locations for radial basis interpolation: computational aspects*, Rend. Sem. Mat. Torino, 61(3) (2003), 343–358.
24. S. De Marchi, G. Elefante, E, Perracchione, D. Poggiali: *Quadrature at "fake" nodes*, draft.

25. S. De Marchi, W. Erb, F. Marchetti, *Spectral filtering for the reduction of the Gibbs phenomenon for polynomial approximation methods on Lissajous curves with applications in MPI*, Dolomites Res. Notes Approx. **10** (2017), pp. 128–137.

26. S. De Marchi, M. Marchetti, E. Perracchione and D. Poggiali: *Polynomial interpolation via mapped bases without resampling*, J. Comput. Appl. Math. 364 (2020), 112347.

27. S. De Marchi, W. Erb, F. Marchetti, E. Perracchione and M. Rossini: *Shape-driven interpolation with discontinuous kernels: error analysis, edge extraction and application in MPI*, SIAM J. Sci. Comput. 42(2) (2020), B472-B491.

28. S. De Marchi, M. Marchetti and E. Perracchione: *Jumping with Variably Scaled Discontinuous Kernels (VSDK)*, Bit Numerical Mathematics (2019). https://doi.org/10.1007/s10543-019-00786-z

29. S. De Marchi, F. Marchetti, E, Perracchione, D. Poggiali: *Python code for Fake Nodes interpolation approach*, https://github.com/pog87/FakeNodes.

30. Stefano De Marchi, Gabriele Santin: *A new stable basis for radial basis function interpolation*, J. Comput. Appl. Math. **253** (2013), 1–13.

31. W. Erb, C. Kaethner, M. Ahlborg, T.M. Buzug: *Bivariate Lagrange interpolation at the node points of non-degenerate Lissajous nodes*, Numer. Math. **133**, 1 (2016), 685–705.

32. W. Erb, *Bivariate Lagrange interpolation at the node points of Lissajous curves - the degenerate case*, Appl. Math. Comput. **289** (2016), 409–425.

33. W. Erb, C. Kaethner, P. Dencker, M. Ahlborg: *A survey on bivariate Lagrange interpolation on Lissajous nodes*,Dolomites Res. Notes Approx.8 (Special issue) (2015), 23–36.

34. G.E. Fasshauer, M.J. McCourt, *Kernel-based Approximation Methods Using* Matlab, World Scientific, Singapore, 2015.

35. M. S. Floater and K. Hormann: *Barycentric rational interpolation with no poles and high rates of approximation*. Numer. Math., 107(2) (2007), 315–331.

36. B. Fornberg, N. Flyer: *The Gibbs phenomenon for radial basis functions*, in *The Gibbs Phenomenon in Various Representations and Applications*. Sampling Publishing, Potsdam, NY, 2008.

37. Bengt Fornberg, Elisabeth Larsson and Natasha Flyer: *Stable Computations with Gaussian Radial Basis Functions*, SIAM J. Sci. Comput. 33(2) (2011), 869–892.

38. B. Fornberg, G. Wright: *Stable computation of multiquadrics interpolants for all values of the shape parameter*, Comput. Math. Appl. **47** (2004), 497–523.

39. B. Gleich, J. Weizenecker: *Tomographic imaging using the nonlinear response of magnetic particles*, Nature 435(2005), 1214–1217.

40. D. Gottlieb, C.W. Shu: *On the Gibbs phenomenon and its resolution*, SIAM Review 39 (1997), 644–668.

41. N. Hale, L.N. Trefethen, *New quadrature formulas from conformal maps*, SIAM J. Numer. Anal. **46** (2008), 930–948.

42. Hewitt, Edwin; Hewitt, Robert E.: *The Gibbs-Wilbraham phenomenon: An episode in Fourier analysis*. Archive for History of Exact Sciences. 21 (2) (1979), 129–160.

43. Nicholas J. Higham: *The numerical stability of barycentric Lagrange interpolation*, IMA Journal of Numerical Analysis **24** (2000), 547–556.

44. A. Jerri: *The Gibbs Phenomenon in Fourier Analysis, Splines and Wavelet Approximations*, Kluwer Academic Publishers, Dordrecht, Boston, London (1998).

45. J.-H. Jung: *A note on the Gibbs phenomenon with multiquadric radial basis functions*, Appl. Num. Math. 57 (2007), pp. 213–219.

46. C. Kaethner, W. Erb, M. Ahlborg, P. Szwargulski, T. Knopp, T.M. Buzug: *Non-Equispaced System Matrix Acquisition for Magnetic Particle Imaging based on Lissajous Node Points*, IEEE Trans. Med. Imag. 35, 11 (2016), 2476–2485.

47. T. Knopp, T. M. Buzug: Magnetic Particle Imaging, Springer-Verlag, Berlin, 2012.

48. T. Knopp, S. Biederer, T. Sattel, J. Weizenecker, B. Gleich, J. Borgert, T.M. Buzug, *Trajectory analysis for magnetic particle imaging*, Phys. Med. Biol. **54** (2009), 385–397.

49. D. Kosloff, H. Tal-Ezer, *A modified Chebyshev pseudospectral method with an $O(N^{-1})$ time step restriction*, J. Comput. Phys. 104 (1993), 457–469.

50. E. Meijering: *A Chronology of Interpolation: From Ancient Astronomy to Modern Signal and Image Processing*, Proc. IEEE 90(3) (2002), 319–342.

51. Yuji Nakatsukasa, Olivier Sète, Lloyd N. Trefethen: *The AAA algorithm for rational approximation*, SIAM J. Sci. Comput. 40-3 (2018), A1494-A1522

52. S. Rippa: *An algorithm for selecting a good value for the parameter c in radial basis function interpolation*, Adv. Comput. Math. **11** (1999), 193–210.

53. T.J. Rivlin, *An Introduction to the Approximation of Functions*, Dover Publications, New York, 1969.

54. L. Romani, M. Rossini, D. Schenone: *Edge detection methods based on RBF interpolation*, J. Comput. Appl. Math. 349 (2019), 532–547.

55. M. Rossini: *Interpolating functions with gradient discontinuities via variably scaled kernels*, Dolom. Res. Notes Approx. 11 (2018), 3–14.

56. Runge, Carl: *Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten*, Zeitschrift für Mathematik und Physik, 46 (1901), 224–243.

57. S. A. Sarra: *Digital total variation filtering as postprocessing for radial basis function approximation methods*, Comput. Math. Appl. 52 (2006), 1119–1130.

58. R. Schaback: *Error estimates and condition numbers for radial basis function interpolation*, Adv. Comput. Math. **3** (1995), 251–264.

59. A. Takaki, T. Soma, A. Kojima, K. Asao, S. Kamada, M. Matsumoto, K. Murase: *Improvement of image quality using interpolated projection data estimation method in SPECT*, Ann. Nucl. Med. 23, 7 (2009), 617–626.

60. H. Wendland: *Scattered Data Approximation*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2005.

61. https://en.wikipedia.org/wiki/Runge%27s_phenomenon#S-Runge_algorithm_without_resampling